T/EP 00/03819

FEDERAL REPUBLIC OF GERMANY

REC'D 02 AUG 2000
WIPO PCT

[handwritten:]
 # 2
EP00/3819

EJU

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

Certificate of Priority for the Filing of a Patent Application

Filing No.:	199 26 206.3
Application date:	June 9, 1999
Applicant/holder:	DaimlerChrysler AG, Stuttgart, Germany
Title:	Configuration system for vehicle electrical system
IPC:	B 60 R and H 04 L

The attached documents are a correct and exact copy of the documents of this application as originally filed.

Munich, July 3, 2000
German Patent and Trademark Office
The President
By order:

[signature]

Nietiedt

[illegible]

DaimlerChrysler AG FTP/E Dr.EW/sn

Stuttgart

Configuration system for vehicle electrical system

The invention relates to a configuration system for a vehicle electrical system to automatically configure the vehicle electrical installations, comprising hardware components, at least some of which are connected to a data bus network, and software components which are implemented in at least some of the hardware components to execute associated functions. The term hardware components as used herein should be understood especially as the various control units (often abbreviated as ECO), including the associated parts and peripherals, such as connecting cables, data buses, connections, sensors and actuators. Each software component that performs an associated function may be implemented in one or in a distributed manner in several hardware components.

Systems for automatically configuring a technical system are intended to support the construction of the system by determining the most optimal system configuration for specified criteria within the limits of the existing capabilities and the given boundary conditions, preferably using a knowledge-based method where the entire configuration-relevant knowledge is stored in a corresponding knowledge-base database. Systems of this type have been proposed, for example, for configuring PC systems and telecommunication systems. With regard to the latter, see Uellner et al., Kundenspezifische Konfiguration von Telekommunikationssystemen [Custom-tailored configuration of

telecommunication systems] - KIKon Project, Deutsche Telekom AG, Darmstadt Technology Center, 1997. With regard to the general fundamentals of knowledge-based configuration, see, for example, A. Günter (editor), Wissensbasiertes Konfigurieren - Ergebnisse aus dem Projekt PROKON [Knowledge-based configuration - results of the PROKON Project, Infix-Verlag, St. Augustin 1995.

In modern automotive vehicles, the electrical/electronic share, i.e., the share of electrical and electronic components forming the vehicle electrical system continues to increase. The wide variety of vehicle electrical systems, a term used herein for convenience to denote the electrical and electronic components as a whole, is a function of, *inter alia*, different automobile manufacturers, different operating regulations in the various countries and different series and equipment variants of the same manufacturer. Hence, there is a need for a configuration system for a vehicle electrical system which can be used to largely automatically configure custom-tailored vehicle electrical installations within the limits of the given conditions and with the highest possible degree of automation, and in which the current configuration, i.e., the actual configuration, can be easily accessed and displayed, for example for service personnel. In addition, there is a desire to be able to reconfigure the respective actual configuration consistently and largely automatically if system components have to be replaced or new components for new functionalities added.

The basic features of a configuration system for a vehicle electrical system of the aforementioned type are described in I. Kreuz et al., Intelligent Configuring System, Proceedings of the 31st ISATA - Automotive Electronics and New Products, Düsseldorf, Germany, June 2 to 5, 1998. The configuration process chain disclosed in that document includes a set of possible configurations specified by development and a set of allowable

configurations derived therefrom, which take into account the restrictions imposed by marketing strategies and which can be used to obtain a custom-tailored, ordered configuration, which in turn is used by production to determine the final actual configuration. All of these configurations are retrievably stored in a database that acts as a central knowledge base. The information on the actual configuration makes it easier for the service personnel to determine the components—a collective term used here to describe both hardware and software components—which make up a given vehicle electrical system. They can furthermore be used to generate individual documentation of the respective actual configuration for the service personnel and to prepare an individual manual.

The technical object of the invention is to provide a configuration system for a vehicle electrical system of the initially described type which enables, in particular, a continuous and relatively simple overview of the corresponding actual configuration and/or the user-friendly and largely automatic configuration of a vehicle electrical system which satisfies the requirements specified by the customer.

According to the invention this object is attained by a configuration system for a vehicle electrical system having the features of claims 1 and 5, respectively.

The configuration system according to claim 1 is characterized by an onboard central actual configuration data memory. An actual configuration data record that characterizes the actual configuration is stored in this memory, so that by connecting a data retrieving component, an overview of the entire current configuration of the vehicle electrical system can be easily obtained in the vehicle itself without the need to access, for example, the database of the automobile manufacturer or various memory components contained in the different hardware

components of the vehicle, which only contain data about the associated hardware component. The actual configuration data memory is either directly or indirectly communicatively connected to all the installed hardware components. This is a prerequisite for relatively simple matching of the actual configuration data stored therein with the current actual configuration, or for keeping the two in agreement, for example if there is an accidental data loss in the actual configuration data memory. Knowing the exact actual configuration is particularly useful for servicing and diagnosing the vehicle electrical system, replacing and repairing individual components, determining the current value particularly for used vehicles and preparing individual manuals for the respective vehicle. Central actual configuration data management ensures high availability and makes it relatively simple to update the data.

In a further refinement of this configuration system according to claim 2, the actual configuration data memory is a flash memory component of a control unit component of the vehicle electrical system. This special control unit component simultaneously acts as a gateway of the vehicle electrical system to an offboard system, which may for example include browser means to graphically represent the entire actual configuration or parts thereof.

In a further refinement of the configuration system according to claim 3, the actual configuration data are stored in an XML file format in the actual configuration data memory. In addition, data relating to the selected structure or grammar of this file format are stored in an associated document type definition file. This makes the actual configuration data memory "self-documenting" in the sense that the actual configuration stored therein can be retrieved and displayed even after a long

time and even if, for example, an offboard browser originally used to display the data is no longer available.

A configuration system further refined according to claim 4 has browser means to display the actual configuration data in a tree structure representation, a function representation and/or a topology representation depending on the system layout and, optionally, one or more additional methods of representation, enabling different views of the actual configuration.

The configuration system for a vehicle electrical system according to claim 5 is characterized by a topology configuration subsystem, a hardware component configuration subsystem, a vehicle configuration subsystem and a graphic user interface to selectively activate the respective subsystem and provide menu-driven user guidance during the activity of the respective subsystem. The topology configuration subsystem is used to enter data for the usable types of hardware components and their data network connection into the configuration system. The hardware component configuration subsystem is used in the development phase of a corresponding hardware component type to select allowable, or develop new, hardware and software modules or components. Upon activation of the vehicle configuration subsystem, the system then largely or fully automatically configures an optimal vehicle electrical system for the vehicle desired by a customer based on corresponding targets specified by the customer, using the information of the topology configuration subsystem and the hardware component configuration subsystem. Any "configuration engine" conventionally used for configuring other electronic systems may be used for this purpose.

In a system further refined according to claim 6, reconfiguration means are provided for computer-aided automatic reconfiguration of a respective vehicle electrical system if a hardware or software component is replaced by a new and different

type which includes at least the functionality of the replaced component. Such a reconfiguration may also be done if an additional functionality that did not previously exist in the vehicle system is added.

A configuration system further refined according to claim 7, provides means for knowledge growing old which assign the stored configuration data a degree of current relevancy as a function of their age and the frequency with which the configuration is used and remove configuration data from the valid configuration data record if their degree of current relevancy has fallen below a specified threshold value. In addition or as an alternative, the reconfiguration means, if any, may use this assignment of a degree of current relevancy if there are several possible, competing components, configuration strategies and/or component relations, so that those with the highest degree of current relevancy are used first and those with a lower degree of current relevancy only in case of conflict.

Preferred embodiments of the invention will now be described with reference to the drawings, in which:

FIG 1 shows a block diagram of a process chain during automatic configuration of a vehicle electrical system using a configuration system for a vehicle electrical system,

FIG 2 shows an excerpt of a block diagram of an automatically configured vehicle electrical system with a plurality of data buses and control units and a central actual configuration data memory,

FIG 3 shows a schematic block diagram of the data structure for the actual configuration data memory illustrated in FIG 2,

FIG 4 is an exemplary detail of a graphic rendering of a tree structure representation of an actual configuration stored in the actual configuration data memory,

FIG 5 is a graphic rendering of an actual configuration in a topology representation,

FIG 6 is a schematic input mask of a graphic user interface for menu-driven user guidance during a configuration process,

FIG 7 shows a screen view provided by the graphic user interface during a topology configuration process,

FIG 8 shows a screen view provided by a graphic user interface to start a vehicle configuration process and select various display functionalities,

FIG 9 is a detail of an actual configuration view in a topology representation as provided by the graphic user interface,

FIG 10 shows a diagnostic model view from the actual configuration data as provided by the graphic user interface, and

FIG 11 shows a schematic block diagram of a reconfiguration process carried out by the configuration system.

FIG 1 shows the configuration process chain during vehicle production for a configuration system for a vehicle electrical system to automatically configure vehicle electrical installations. As illustrated in the figure, the present application case of a knowledge-based configuration system involves the typical process sequence described below.

The system developers produce various hardware and software components to satisfy vehicle-specific functionalities, which

together with their associated component relations, i.e., the existing rules and boundary conditions, form a set C_P of possible configurations. To this end, all subsystems of the vehicle electrical system, such as control units, actuators, sensors, etc., are recorded at a central location. This also includes referencing externally available documents, such as information from manuals or diagnostic modules for a control unit function. For this purpose, the development departments have interfaces among the existing systems to record and release systems for central data management. Any missing information or references are added.

However, because of marketing strategies or country-specific regulations, for example, not all possible configurations are meaningful and allowable. Thus, a remaining set C_a of allowable released configurations, which takes into account such conditions, is generated from the possible configurations. For example, the rules and/or boundary conditions regarding the installability of various electrical/electronic components are recorded in the development and marketing departments. These rules and/or boundary conditions can be, for example, technical restrictions to ensure compatibility among certain control units or restrictions imposed by market strategies based on which a technically feasible function may be used only with certain minimum features, such as motorization. The set of technically possible configurations C_p and the set of released configurations C_a is thus defined by development and marketing through basic specifications, i.e., the planned functionalities, the networking topology, the released control units and the buildability rules, i.e., the component relations and boundary conditions. These configuration quantities thus contain not only technical information but, through rules and external references, the entire expert knowledge regarding the buildable products.

Sales, in turn, using computer-aided automatic configuration, takes the released configurations C_a to derive the optimal ordered configuration c_o in response to the customer's request for a desired, ordered vehicle. Criteria for an optimal configuration are, for example, costs, data bus communication and of course absence of conflicts in the configured system. During production, the ordered configuration c_o is used to generate the actual configuration c_e of the vehicle electrical system as the vehicle is assembled from the separate components. For this purpose, particularly the suitable hardware components, i.e., primarily the various control units, are selected and the required software is implemented, the necessary software modules are appropriately combined during configuration and preferably transmitted to the vehicle by means of flash software. All these configurations or configuration records are retrievably stored in a central database Db. The actual, largely automatically determined configuration c_e is then available to service personnel, for example.

Sales and logistics use the set of released configuration c_a directly as a knowledge base for a knowledge-based configuration system, with which, for example, information to generate a bill of materials for production, a communication matrix of the control units connected to the data buses, a general circuit diagram from the topology information and the individual circuit diagrams of the control units, an individual user manual and individual diagnostic information from the diagnostic modules for the control units can be generated by automatic configuration. The configuration target used by the configuration system is the customer's requirements regarding the vehicle, i.e., the desired scope of functions. A parameterization of the sales systems from the knowledge databases, i.e., of the number of allowed configurations c_a may further be provided, in which the marketing departments also supply information about each function or each

equipment feature for the sales brochures, so that customized sales brochures may be generated automatically as a function of the customer profile. In this case, the system is not limited to the electrical system of the vehicle.

The aforementioned bill of materials also includes information about programming the control units which is required for production, i.e., the software modules required for production are identified during configuration and made available to production via references. The bill of materials is the essential component of the ordered configuration c_o. It may still give production a certain degree of freedom to increase deliverability. For example, production may be given discretion to install some freely programmable controllers rather than several special control units. This makes it possible to implement desired functions in a distributed manner by programming with software modules which are identified by the configuration system. Even if this procedure is more costly, it may still be useful in a specific case, e.g., to allow the vehicle to be produced more quickly if there is a delivery bottleneck for a special control unit.

Characteristically, the actual configuration c_e of the vehicle electrical system is stored onboard in the ordered vehicle F. Onboard storage of the actual configuration makes it easier not only to perform diagnostic and maintenance work but also to implement new (software) functionalities, automate software updating and generate vehicle-specific manuals. It may also be used to determine the value of the used vehicle at any time. The used car dealer can transfer the actual configuration data from the onboard actual configuration data memory and determine the resale value based on the features of the vehicle. The onboard actual configuration data is usually stored at a central location in the vehicle. In addition to the hardware

components ECU1, ECU2,…, a central control unit CECU is provided as illustrated in FIG 2, which has a flash memory part or another electronic memory that acts as an actual configuration data memory ECO in which the exact actual configuration, or more precisely the minimum data required to determine the actual configuration, is stored. The actual configuration data thus describe all electrical and electronic hardware and software components of the respective vehicle, including their component relations. Storing the actual configuration onboard the vehicle ensures that the data is current even after several successive system changes, the documentation is inherently vehicle-specific, the actual configuration is available as part of the vehicle, and "invisible" system components, such as software components, are easily identified.

At the same time, the additional central control unit CECU acts as a gateway between the vehicle and the environment of the vehicle, i.e., an external system 1 may be connected to this control unit CECU for data communication with the vehicle. This makes it possible, among other things, to use the external system 1 to retrieve the actual configuration data stored in the actual configuration data memory ECO of the gateway control unit CECU and display it by means of a suitable browser.

To ensure that the actual configuration stored in the actual configuration data memory ECO corresponds in fact to the real, actual configuration, it is provided that each control unit component ECU1, ECU2, … is capable of identifying itself and the central control unit CECU, as illustrated in FIG 2, is directly or indirectly communicatively connected to each of the other control units ECU1, ECU2,…, such that the other control units ECU1, ECU2, … are at least in part conventionally connected to one or more data buses Bus1, Bus2, …. The actual configuration data stored in the actual configuration data memory ECO can thus

be automatically matched to the real system and if necessary brought into agreement again if such agreement was lost, for example because of an accidental data memory failure or damage in the central control unit CECU.

It is particularly advantageous if the onboard documentation made possible by storing the actual configuration onboard the vehicle is self-documenting. In this case, the stored actual configuration can be read out and correctly interpreted even after a long time and even if the offboard system 1 that contained the browser used to display the actual configuration is no longer available. This is achieved if the actual configuration data memory ECO has the actual configuration data in a suitable file format and also has information to define the structure, i.e., the grammar of this file. A suitable file format of this type is the XML (Extensible Markup Language) file format, which was published by W3C (World Wide Web Consortium) and represents a subset of SGML (Standard Generalized Markup Language). The documents are stored hierarchically and are machine-readable. It is possible, for example, to store tree structures and to interpret their contents by computers. Tags are used to separate and identify data fields and thus help make the data self-documenting.

As shown in FIG 3 by way of example for a vehicle with the identification number "12345," the file's structure or grammar information is stored in a DTD (Document Type Definition) file together with the actual configuration data in the XML format. This ensures consistent, i.e., accurate, actual configuration documentation even after years, so that the actual configuration data can be permanently interpreted and also modified.

Various options are available to represent the actual configuration depending on the system design. A first option is to render it in a tree structure representation as partly

illustrated in FIG 4 with a few main branches. As may be seen, the actual configuration data in this case include vehicle identification information, information on the vehicle owner, information on technical data, such as performance, vehicle dimensions, etc., which are useful in the event of resale, maintenance data on the last oil change, for example, and other data relating to the actual vehicle electrical system regarding existing data buses and existing hardware and software components. The data on the hardware components include not only their internal details and the implemented software, but also connectable actuators and sensors and available and used resources, such as connecting cables, connections, bus identifiers, memory areas and CPU use. Based on this information, hardware and software replacement parts can be checked for compatibility with the rest of the system if a defective component must be exchanged. By documenting the functions of the software modules of a control unit, the function of the control unit itself is also documented. Thus, the stored actual configuration data contain all the information required to find all control units and software modules involved in the embodiment of a given functionality. If a certain functionality is no longer available, a service department can query a browser for the actual configuration data memory ECO to show all the components that might be the cause.

Other possibilities to represent the actual configuration are the function representation, which can be relatively easily realized by sorting all the components by their function and forming a corresponding new tree, such that the branches closest to the roots represent the functions and the adjacent branches the hardware and software components to implement the respective function, and the topology representation, an example of which is illustrated in FIG 5. In such a topology representation, the control units connected to data buses are a good starting point

to navigate through the stored actual configuration data. The information for this representation can be found by sorting the control units according to their data bus connection, such that control units which are connected to more than one data bus represent gateways. The display of this topology representation is more difficult than a tree structure display, however. An exponentially increasing number of computation steps is required if the data buses are not to intersect, and the representation is often no longer easy to understand. To make it easier to display this representation in the present case, certain additional information is therefore preferably stored in the actual configuration data memory ECO.

To this end, a diagram of the topologies for a fully equipped vehicle conventionally drafted during the development of the vehicle is divided into rectangles of equal dimensions, each of which contains no more than one data bus and is identified by coordinates. The course of a data bus can then be stored in the original diagram as coordinates of the squares that contain parts of this data bus. If the topology of the individual vehicle is drafted, the browser can use the coordinate information stored for the various data buses to position the buses. Gateway control units are located at the edges between two adjacent squares, respectively. This makes it possible to achieve a clear topology representation as shown in FIG 5 with relatively little computation effort. By clicking on a respective ECU in this representation, the user can go to the associated software, the associated connections, etc. The aforementioned representation types and any others that may be implemented are preferably encoded directly in the associated browser. The use of XSL (Extensible Style Sheet Language) format, which is to be provided by W3C in the near future, appears interesting as a future development. This language helps define representations of XML data which are used in a general XML browser.

FIG 6 to 10 use schematic and actual screens by way of example to illustrate user-friendly menu-driven guidance through the vehicle electrical configuration system by means of a suitable graphic user interface. FIG 6 schematically shows four buttons of an input mask. A first button B1 gives access to a topology configuration, a second button B2 to a control unit configuration, a third button B3 to a vehicle configuration and a fourth button B4 to a vehicle reconfiguration.

Clicking on the first button B1 activates a topology configuration during which basic specifications can be entered in the respective windows regarding the types of control devices that are installed and how they are networked. This assists the developer by capturing experimental values. For example, the corresponding topology configuration subsystem can initially automatically position components on those data buses where they have previously been disposed most often. FIG 7 shows a typical actual screen during topology configuration. In the left part of the window, the possible control unit types are listed for selection, whereas in the subsystem on the right, previously selected control unit types are shown with their automatic placement on the corresponding data bus. Shown are, for example, an electrohydraulic brake EHB on a chassis bus, an engine control unit MS, a proximity controlled cruise control ART and an active body control ABC on an engine bus, various components relating to the passenger compartment on a passenger compartment bus, such as an instrument cluster, an airbag control unit, automatic headlight leveling, a roof operating unit, a signal detection and control module, a voice-activated control system, various components on a KIN/telematics bus, such as radio, CD player and navigation unit, and a pneumatic control unit and trailer hitching unit on an accessory bus.

In a next step of the menu-driven configuration, development selects control unit configuration to release the concrete allowable control units for the various control unit types, or designs them if they do not exist yet. For each control unit type, allowable hardware and software modules are selected or newly defined, e.g., a suitable engine control unit with suitable control software. Characteristics and references to the associated documentation types are stored for each individual control unit and the associated software.

By clicking on the vehicle configuration button B3, an actual configuration can be automatically generated, i.e., a computer-aided automatic configuration of the vehicle and its electrical system components based on the information from the topology configuration subsystem and the control unit configuration subsystem taking into account the targets specified by the customer. A corresponding menu is shown schematically in FIG 8. A configuration process is started by clicking on a configuration start button B5 located in the center column of the menu. In the left menu column B6, all the functionalities desired by the customer can be selected. In the lower part 2 of the center column, the vehicle identification number can be entered along with the name and address of the vehicle owner. The right menu column contains a number of display request buttons for selecting specific configuration and display functionalities.

For example, a bill of materials button B7 can be activated to generate a bill of materials for the actual configuration determined. A topology button B8 can be used to display the actual configuration in a topology representation. A detail of such a representation is shown in FIG 9. As may be seen, this representation basically corresponds to that of the topology configuration illustrated in FIG 7, but here the figure shows the individual, specific control units selected for the determined

actual configuration while FIG 7 shows the respective control unit type.

With a flash software button B9, the required software modules can be combined during configuration and downloaded into the vehicle using conventional flash software. A diagnostic model button B10 is used to activate a diagnostic model function to assemble the referenced diagnostic modules into an individual diagnostic model tailored for the respective vehicle and to display this individual diagnostic model. FIG 10 shows a detail of a typical diagnostic model representation as generated by the configuration system.

Furthermore, a communication matrix button B11 activates the display of the data bus identifiers used, e.g., CAN identifiers, as a communication matrix. In addition, the actual configuration also contains information about all other resources of the control units used, such as memory, CPU capacity, I/O ports, etc. A manual-generating button B12 provides an additional functionality to automatically produce a manual. It is used to generate a vehicle-specific manual by combining the parts of the manual referenced by the control units. The user thus receives precisely the operating instructions that relate to the components actually installed in the vehicle and is not burdened with superfluous information about components that are not applicable. The resulting manual in electronic form can be easily browsed on screen or printed out if necessary.

The menu prompting described above with reference to FIG 7 to 10 thus makes available a very user-friendly vehicle configuration process controlled by a menu-based graphic user interface. Another function of the configuration system for the vehicle electrical system, which is activated via the input mask illustrated in FIG 6, provides a reconfiguration button B4 to select a new or reconfiguration. This reconfiguration subsystem

is used, in particular, to update the actual configuration after one or more hardware and/or software components have been replaced or added. When a component is replaced, this is important particularly if the same component is unavailable and a different type providing the required functionality must be used instead. This replacement component must satisfy all the dependencies applicable to the respective system component. The reconfiguration subsystem helps find a suitable replacement component. The system configuration knowledge about the modules that are currently installed in the system is provided by the actual configuration data stored in the form of an XML document in the vehicle's actual configuration data memory.

If, for example, the outside mirror on the passenger side was mechanically damaged, the smallest replacement unit is the entire outside mirror unit, including servomotors and position sensors, along with the mechanical mirror element itself. The replacement part is a mechanically identical unit but may for example have a newer version of the position sensors, perhaps one with a higher resolution. In conventional systems, this replacement part could not be used because the software in its control unit does not match the signals of the position sensor. The reconfiguration subsystem offers an improvement here, because it determines in a subsequent step that the software module for controlling the position of this outside mirror needs to be replaced and also indicates other necessary changes. Analogously, updating the system with new functionalities requires new hardware and software modules to be added in such a way that all mutual dependencies are successively satisfied without conflict. With this reconfiguration functionality, the changes required can be identified relatively easily and without errors.

One possible realization of the reconfiguration subsystem consists essentially of two parts. The one part is a knowledge

base that contains all the necessary knowledge about available components and their possible parameters, the dependencies among components, strategies for a successful exchange of parts and information as to which modules are currently in stock. To model this knowledge, an object-oriented method may be used, such as the one described in Günter et al. for the KONWERK Project. The second part is an inference engine that works with this knowledge. The KONWERK kernel may be used for this purpose because it easily allows adding one's own strategies. On the other hand, a resource-based strategy is suitable because the aforementioned component dependencies are chiefly the result of the given resources of the modules and can therefore be adequately modeled (see, for example, Heinrich et al., The Resource-Based Paradigm: Configuring Technical Systems from Modular Components, AAAI-96 Fall Sympos. Series: Configuration, MIT, Cambridge, MA, November 9 to 11, 1996, p. 19).

As described in Crow et al., Model-Based Reconfiguration: Diagnosis and Recovery, Computer Science Laboratory, Menlo Park, CA 94025, USA, NASA Contractor Report 4596, May 1994, reconfiguration can be used to obtain an FDIR (Fault Detection, Isolation and Reconfiguration) system by using diagnosis and reconfiguration in systems with "standby" replacement parts. In the present case, however, no "standby" replacement parts are assumed, and reconfiguration is not limited to parts replacement after diagnosis but may also be used for system updating. A first starting point is a part that needs to be replaced, e.g., the last step in an FDIR process chain. The system user identifies the part, and the reconfiguration system removes it from the current system configuration which is determined from the stored actual configuration. The configuration system then establishes that the system is no longer complete. Completeness in this connection means that the system satisfies the specification given by the set of functions previously contained in the system

and that all dependencies are satisfied. The second starting point concerns the case where a new functionality is to be added. In this case, too, the system is no longer complete because the specification has been changed and consists of the previous functionalities plus the new desired functionality.

In both cases, a composition step of the reconfiguration system consists of an attempt by the system to add a component that provides the missing functionality. For this purpose, it may select the same type of component which the user previously removed if this component is still available and has not already been ordered elsewhere. However, the system can also "decide" to select a newer version of the component, or a completely different component that is currently in stock, depending on the strategy and the optimization rules that are actually used. To satisfy all the dependencies, the reconfiguration must usually be done in steps. A third interesting situation results if the rules/dependencies for the system have changed, e.g., because new dependencies were discovered by the development departments at a later stage and were added to the knowledge base. Even transporting the vehicle to another country with different regulations may be a reason for this. In this case, the reconfiguration system determines whether all the dependencies are still satisfied and initiates a step-by-step reconfiguration if this is not the case.

For the reconfiguration process, there are several conventional options. In Crow et al. cited above, reconfiguration is modeled as an analogy to the model-based diagnosis paradigm as formalized by Reiter (R. Reiter, A Theory of Diagnosis from First Principles, Artificial Intelligence, 32(1), April 1987, p. 57). In the present case, a reconfiguration based on a mechanism that attempts to use as much as possible of the well-researched classical configuration paradigm is preferred. This mechanism

consists of adding components to obtain a desired functionality and of decomposing the system if a conflicting pair is detected. This is similar to the fundamental configuration method with backtracking used in classical configuration systems as described in Günter et al. and Heinrich et al. cited above. The difference relative to normal backtracking is that not only those parts that were previously added by the algorithm but also those that were incorporated into the system before the algorithm was started can be removed.

The algorithm has two distinguishable phases, the composition phase and the backtracking or decomposition phase. In the composition phase, the reconfiguration system tries to find a component that provides at least one of the desired functionalities. If one or more desired functionalities are missing, a decision must be made as to which component should be added next. This can be, for example, the component with the highest number of missing functions or with the lowest number of conflicting pairs. This phase ends when conflicting pairs occur or the system is complete with regard to a given specification. In the decomposition phase, the system removes all components that are part of a conflicting pair. Since adding a component in the composition phase can result in several conflicting pairs, the decomposition phase basically consists of a loop in which the components belonging to at least one conflicting pair are identified and the component to be removed next according to a given optimization strategy is selected and then removed, such that the components contained in the greatest number of conflicting pairs can be removed first, for example. The decomposition phase ends when there are no more conflicting pairs. The composition and decomposition phases are alternately executed in a loop cycle until the system is complete during the composition phase with regard to a given specification.

Since it is common to develop different versions of hardware and/or software components, especially also during the life of a vehicle, it is useful if the reconfiguration system is able to deal with the problem of different component versions in two ways. On the one hand, it can check whether a conflict of any kind occurs when a newer version is used and also help resolve the conflicts by finding alternative components having the same functionality. On the other hand, it can identify all the parts that are to be replaced by a new version, even if they are still functioning properly, if a condition is added to the knowledge base which says that the older version part itself represents a conflict. The reconfiguration algorithm satisfies these two aspects without any additional programming effort. The new version component must be added to the knowledge base as an available part, and the older version must either be removed from the knowledge base or marked as being no longer available.

FIG 11 graphically illustrates the reconfiguration function by symbolically representing the system components as puzzle pieces of the respective vehicle F. It is assumed, by way of example, that a certain functionality or component K of the existing actual configuration needs to be replaced. The knowledge-based configuration system uses its reconfiguration subsystem to determine not only a new component Kn for the component K, which needs to be replaced in any case, but also other new components K1n, K2n to replace corresponding current components K1, K2, until the system is complete again and free of conflicts. By replacing the current components with the automatically determined new components and optionally adding additional components, a newly configured vehicle Fn with the desired functionalities is obtained.

Preferably, the present configuration system for a vehicle electrical system takes into account an aging aspect of the

configuration or reconfiguration process. This takes into consideration the fact that over the years not only the component versions but also the knowledge about dependencies or (re)configuration strategies change. Taking into account knowledge that has grown old can improve the reconfiguration process, so that both the components and the knowledge can change continuously and automatically and not just abruptly with the switch to a new version.

One advantage of using knowledge grown old is that it improves the knowledge base by automatically removing knowledge that has not been used for a prolonged period. This keeps the knowledge base sufficiently small and the associated configuration engine correspondingly fast. In special cases where old knowledge is required, for example to reconstruct an oldtimer vehicle, it may be useful to use backup versions of the knowledge base which make it possible to go back to any desired earlier date.

A second essential reason for determining knowledge grown old is that the configuration process is accelerated as a result. It is highly probable that knowledge that has often helped find a solution and is relatively current is most likely to solve a current problem, so that it is plausible to use this knowledge first, which is therefore termed current knowledge. This means that to accelerate the configuration or reconfiguration process, current components and strategies are given preference and current dependencies are checked first.

In the present case, to classify the age of knowledge, a parameter act which indicates the degree of current relevancy and which is used in the form act=usef/age is defined as the quotient of a parameter usef, which indicates how often the respective knowledge has led to a solution, or a component has been part of a solution, and a parameter age, which indicates the age of the

knowledge or component relative to the date when it was initially stored in the knowledge base. If a component's degree of current relevancy act continues to drop toward zero over a prolonged period of time, the component can be removed from the currently valid knowledge base when it falls below a specified threshold value.

Preferably, the utilization level parameter usef is weighted relative to the age of the corresponding component, such that the degree of current relevancy act is then determined from the modified relation $act=usef^{c}/age$, where the exponent c can be defined as appropriate. Specifically, to emphasize age, it is set at a value between zero and one and to emphasize the utilization level it is set at a value greater than one. To calculate the degree of current relevancy act, the parameters usef and "birth date" are stored in the knowledge base for all knowledge and for all components. The exponent c is valid for the entire system and can be redefined for each reconfiguration process, enabling the system user to define the relevancy of age.

DaimlerChrysler AG
Stuttgart

FTP/E Dr.EW/sn

Claims

1. Configuration system for a vehicle electrical system to automatically configure vehicle electrical installations, comprising hardware components (ECU1, ECU2, ...) at least some of which are connected to a data bus network (Bus1, Bus2, ...), and software components implemented in at least some of the hardware components to execute associated functionalities,
characterized by
an onboard central actual configuration data memory (ECO) to retrievably store an actual configuration data record characterizing the actual configuration of the respective vehicle electrical system, such that the actual configuration data memory is directly or indirectly communicatively connected to all the hardware components (ECU1, ECU2, ...).

2. Configuration system for a vehicle electrical system as claimed in claim 1, further
characterized in that
the actual configuration data memory (ECO) is formed by a flash memory part of a control unit component (CECU) which acts as a gateway of the vehicle electrical system to a connectable offboard system (1).

3. Configuration system for a vehicle electrical system as claimed in claim 1 or 2, further
characterized in that
the actual configuration data are stored in an XML file format in the actual configuration data memory (ECO) and data about the

structure of the XML file format are stored in an associated document type definition file (DTD).

4. Configuration system for a vehicle electrical system as claimed in any one of claims 1 to 3, further characterized by
browser means to render the actual configuration data in a tree structure representation, a function representation and/or a topology representation.

5. Configuration system for a vehicle electrical system to automatically configure vehicle electrical installations, comprising hardware components (ECU1, ECU2, ...) at least some of which are connected to a data bus network (Bus1, Bus2, ...), and software components implemented in at least some of the hardware components to execute associated functionalities, particularly as claimed in any one of claims 1 to 4,
characterized in that
- it comprises a topology configuration subsystem (B1) to input data regarding the types of usable hardware components and their data network connection, a hardware component configuration subsystem (B2) to select and/or newly develop hardware components of the respective type, and a vehicle configuration subsystem for computer-aided automatic configuration of a respective vehicle electrical system as a function of specified targets using the topology configuration subsystem and the hardware component configuration subsystem, and
- graphic user interface means are provided for menu-driven user guidance during the activity of the topology configuration subsystem, the hardware component configuration subsystem and the vehicle configuration subsystem.

6. Configuration system for a vehicle electrical system as claimed in any one of claims 1 to 5, further characterized by

reconfiguration means (B4) for computer-aided automatic reconfiguration of a respective vehicle electrical system if at least one component is replaced by at least one new component having a corresponding function but being a different type, or if at least one additional component for a new functionality is added, or if at least one component relation is changed.

7. Configuration system for a vehicle electrical system as claimed in any one of claims 1 to 6, further characterized by
means for knowledge growing old which assign the stored configuration data a degree of current relevancy (act) as a function of its age (age) and configuration use frequency (usef), such that they remove configuration data from the valid configuration data record if their degree of current relevancy has fallen below a specified threshold value, and/or such that, if reconfiguration means are provided, said reconfiguration means, if there are several possible components, configuration strategies and/or component relations, first use those with the highest degree of current relevancy during the reconfiguration.

DaimlerChrysler AG FTP/E Dr.EW/sn
Stuttgart

Abstract

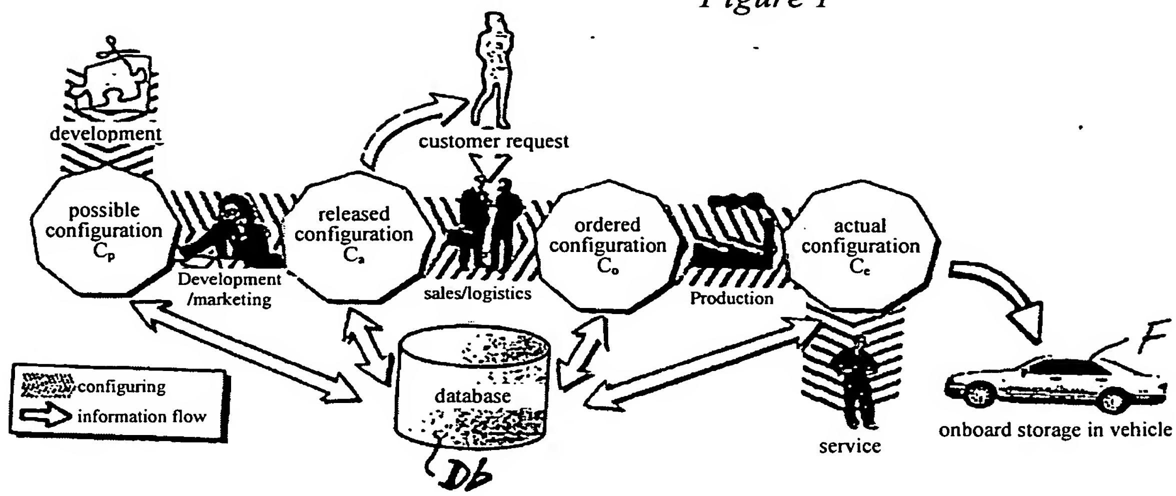
1. Configuration system for vehicle electrical system.

2.1 The invention relates to a configuration system for a vehicle electrical system to automatically configure vehicle electrical installations, comprising hardware components, at least some of which are connected to a data bus network, and software components which are implemented in at least some of the hardware components to execute associated functionalities.

2.2 The invention proposes a central onboard actual configuration data memory for retrievably storing an actual configuration data record characterizing the actual configuration of a respective vehicle electrical system, such that the memory is directly or indirectly communicatively connected to all hardware components. In addition or as an alternative, the system has a topology configuration subsystem, a hardware component configuration subsystem and a vehicle configuration subsystem, including a graphic user interface for menu-driven user guidance.

2.3 Use for automatic custom-tailored configuration of the vehicle electrical system of automobiles, for example.

Figure 1



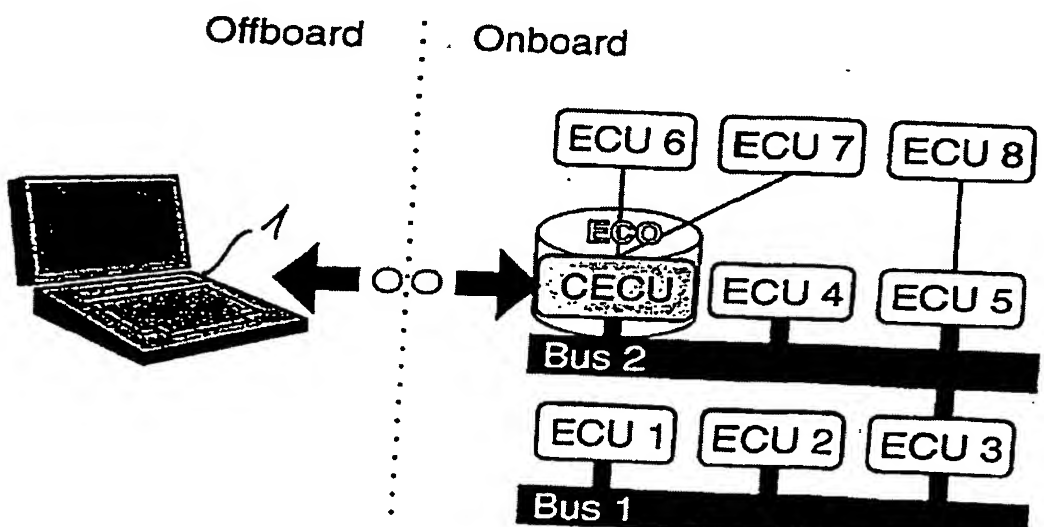
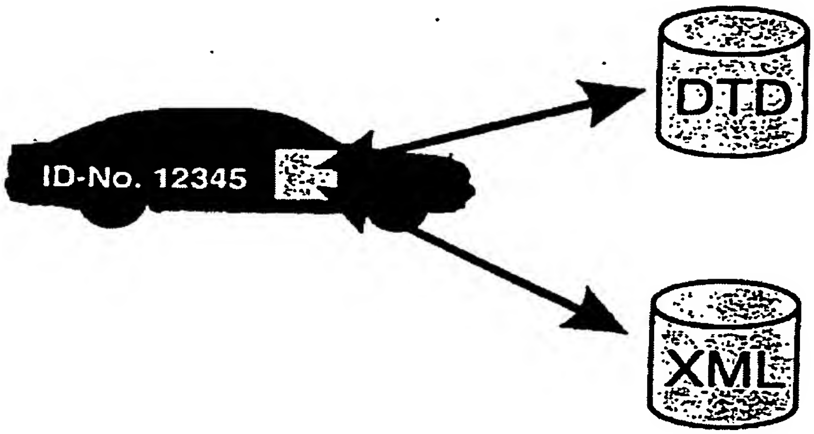


Figure 2

Figure 3



[Note: sensors/actors should probably read sensors/actuators]

Vehicle	
– Identification	i.e. VIN
– Owner	
– Technical Data	for Resale
– Data for Maintenance	i.e. date of last oil change
– Data Busses	i.e. CAN-Busses
– ECU	including all
– loaded Software-Module	- connectors
– loaded Software-Module	- connected busses
...	- sensors/actors
– ECU	- programming ressources
– ...	
...	

Figure 4

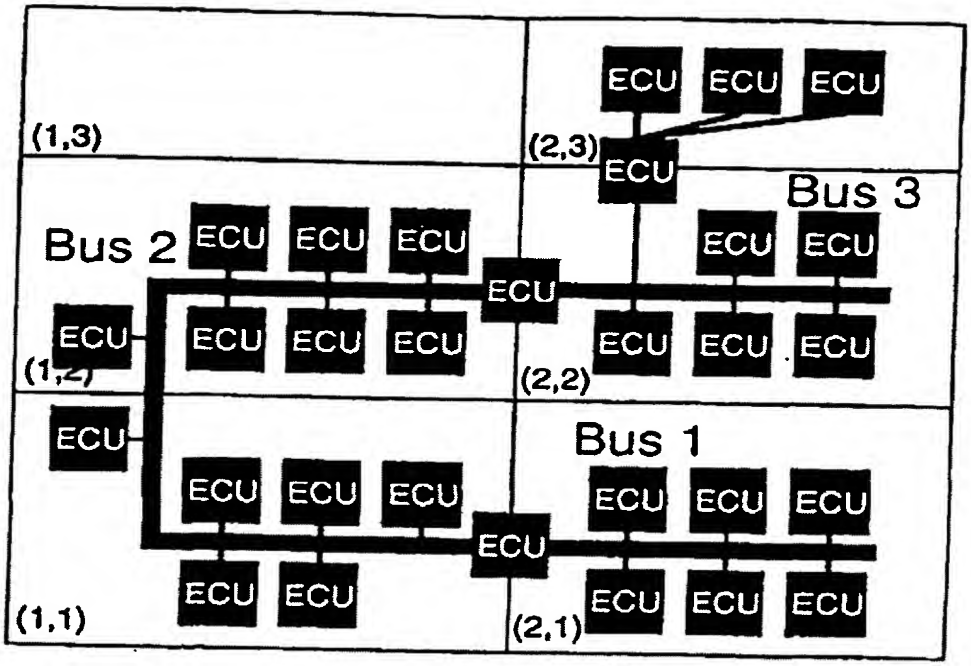


Figure 5

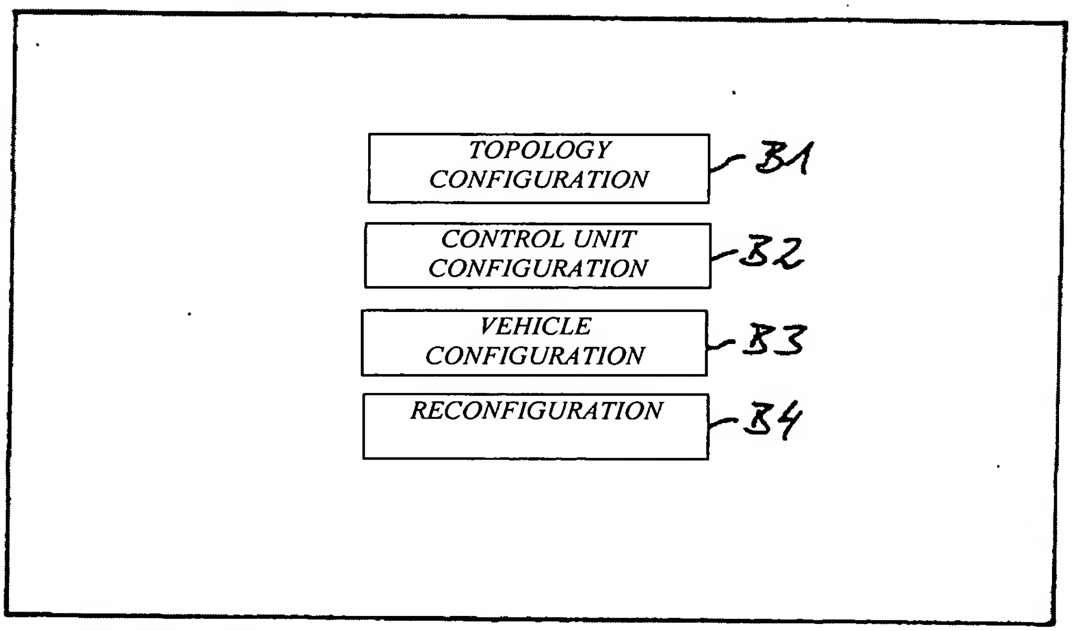


Figure 6

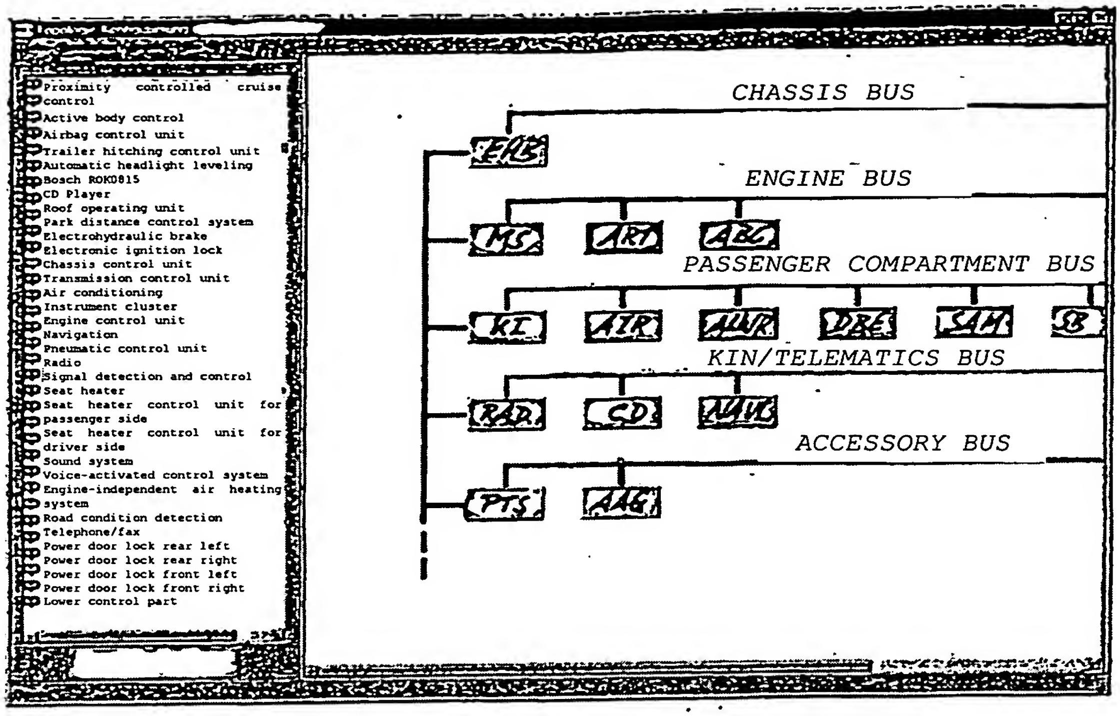
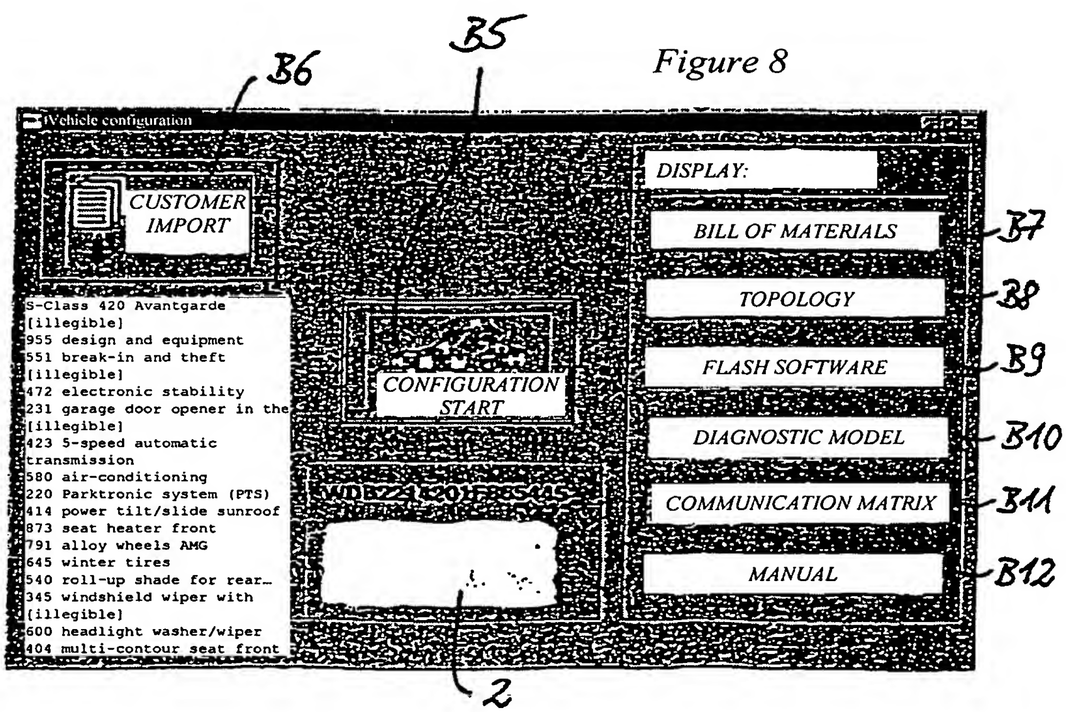


Figure 7

Figure 8



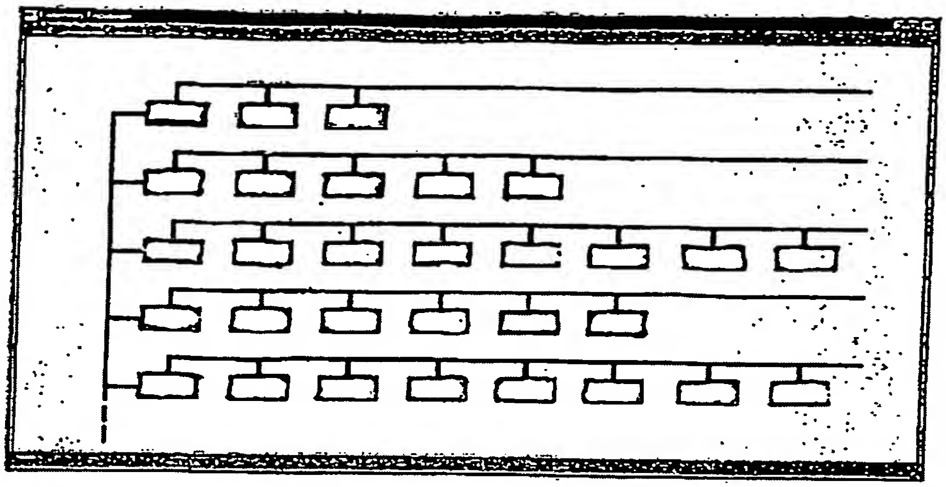


Figure 9

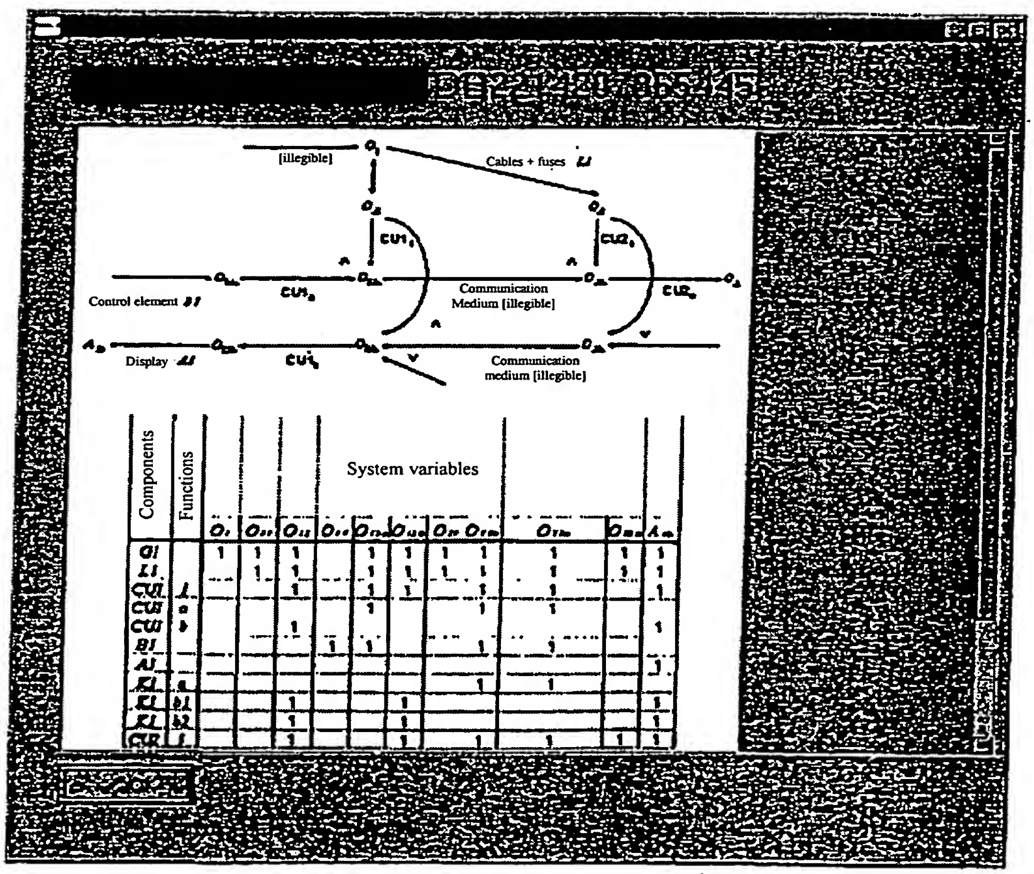


Figure 10

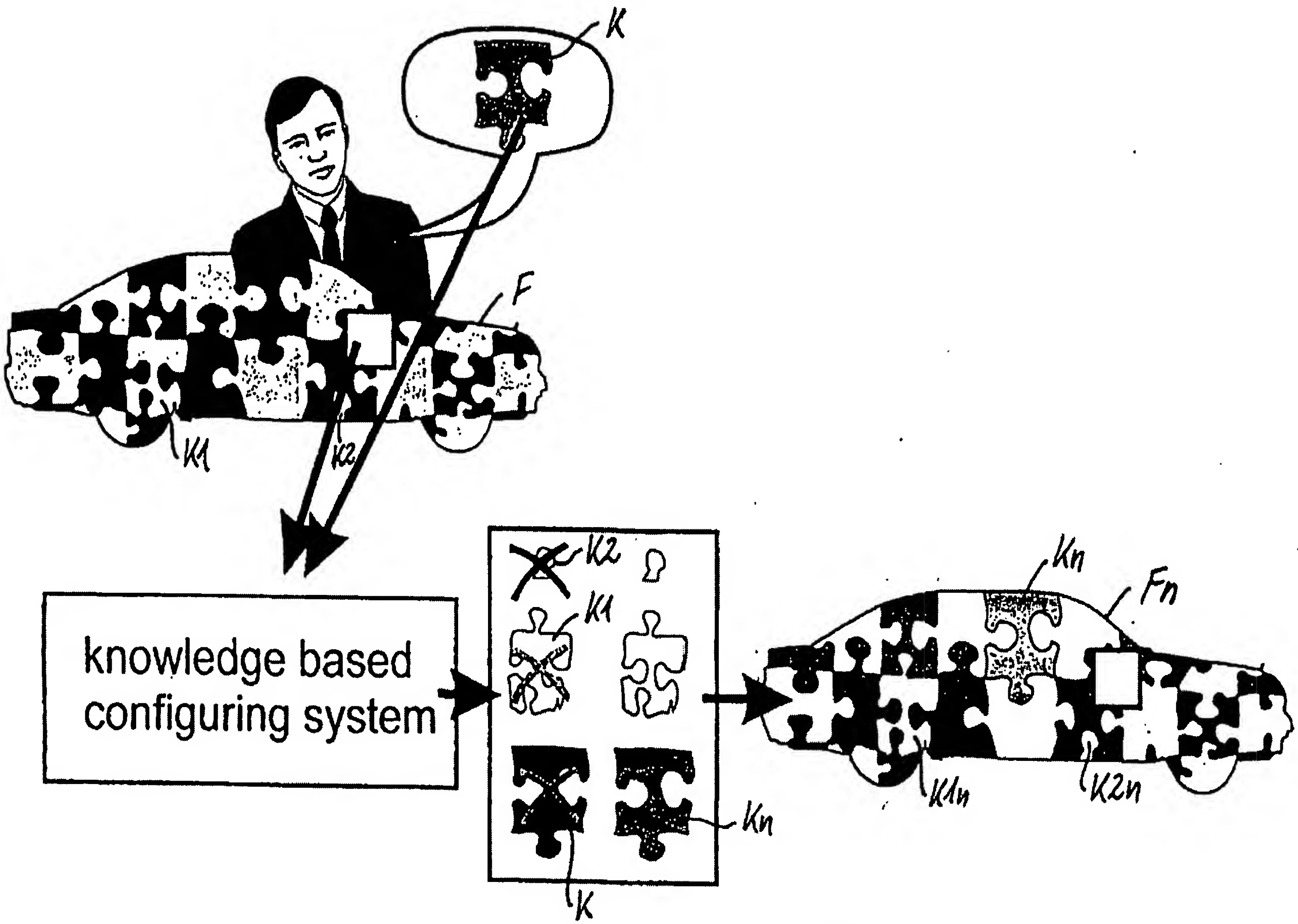

K
F
K1
K2
knowledge based
configuring system
K2
K1
K
Kn
Kn
Fn
K1n
K2n

Figure 11